Document Ref: DMON-BDS
Issue No: 1. July 1986

Greenbank Electronics
Telephone: 051-645 3391

Interak 1

DMON BOOT EPROM

DMON Boot EPROM and Disk Monitor

FEATURES

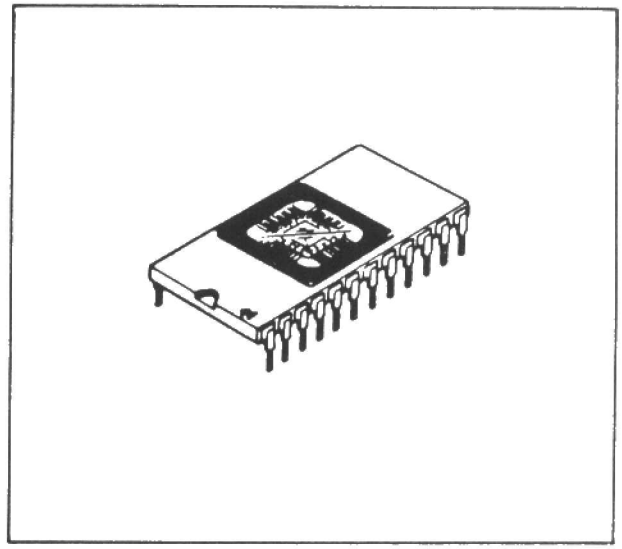
* Fits onto MZB-3 CPU Card at location E000 (hexadecimal).
* Supplied in suitable 5V-only EPROM.
* Boots up all sizes of disk.
* Boots up single and double density Interak format disks.
* Instructions included.
* Numerous relevant and useful commands.
* Primarily to boot CP/M, but also useful for other (eg user-written) programs.
* To run in RAM at 4 MHz with no Wait states.
* Designed for Interak 1 System.
* Low cost.
* Monitor commands to read and write specified numbers of sectors at specified tracks.
* Printer driver included.
* Single Density uses 128 bytes per sector.
* Double Density uses 512 bytes per sector.
* Full 10 double density sectors per track per side (ie not 8 or 9) on 3.5"/5.25" diskettes.
* 15 double density sectors per track per side on 8" diskettes.
* Can cope with transfer rate of 500 kilobits per second (twice as fast as the maximum for many other systems).
* Several versions available to suit differing installations.

DESCRIPTION

There are two fundamental approaches to Interak. One is to build it as a tape based system and the other is to build it as a disk based system. (Although of course, owing to the modular nature of the system the distinctions are neither clear cut nor permanent.) However if you are initially using tape alone, you should not begin with the DMON monitor, since this has no tape routines whatever, but instead consider the use of ZYMON to start with (ZYMON has tape routines but no knowledge of disk).

The "Disk Monitor/Boot" EPROM ("DMON" for short) is capable of stand-alone use. At least three versions of this are available and the customer can choose the appropriate one to suit his system, as follows:

One type of system is the natural development of the "Interak 1", and has a memory-mapped screen (eg VDU-2K, set up for 64 column operation) at location F000H, and a parallel ASCII keyboard interface (eg LKP-1 at port 40H). DMON to suit this is called "DMON-M".



DMON Boot EPROM and Disk Monitor

Surplus

The next type of system is the "Serial" type which has a full 64K of RAM, with a serial combined VDU and Keyboard, (ie a traditional "Terminal" or "Console"). DMON to suit this is called "DMON-S". At present we do not supply such a VDU terminal, but many suitable terminals are available surplus and second-hand from dealers in such equipment.

The last type is called "DMON-H", and can be thought of as a hybrid between the two already mentioned. DMON-H is to suit a standard Interak keyboard arrangement (eg the keyboard interface LKP-1), but this time with a serial type of VDU. Again, we do not at present supply such a VDU but DMON-H is ready for it if we do (or if you have a serial VDU of your own).

Only identified standard hardware configurations will be catered for; it is not planned to offer individually customised versions for individual specific hardware, but of course we would be pleased to receive full information of any particular set up you might have which is not suited by one of the above 3 variants of DMON, so that we could consider introducing an additional standard.

Printer

Whatever type of DMON is in use, the printer and disk drive arrangements are the same. The printer ports are user-selectable, using a parameter in DMON's SP ("Set Parameters") command. Communication with the printer is via two ports, which we call "Status" and "Data". The printer can be either serial (eg RS-232) or parallel (eg "Centronics" interface) because, in the method we use, the same software suits both types.

Selecting the status port automatically defines the data port. (We use the convention that the status port address is an even number, and the data port is the odd number immediately after it. The Z80's ports begin with port number 00.)

A serial interface often uses a "UART" (universal asynchronous receiver transmitter) type of device. The software routine in DMON checks the UART "Status" port to see if the "TBMT" (UART Transmit Buffer Empty) flag

is set, and only sends data, (to the "Data" port), when the transmit buffer is empty. If data were to be sent without this check, ie if a continuous stream of characters were to be sent to the printer at maximum computer speed very few printers could keep up. Despite malicious rumours to the contrary, (spread by non-Z80 users), the Z80 is very fast, and can certainly send data to a typical printer very much faster than it can be printed, so the "TBMT" method is vital to avoid lost characters.

In the case of a parallel printer interface, we naturally do not use the UART, which is a serial device. We do however still keep to the same ports, and continue to refer to them as "Status" and "Data". One of the signals which can be used for "handshaking" in the parallel printer interface is called "Busy". The printer generates this when it is incapable of receiving any more characters for printing. We connect this signal in our parallel printer interface to exactly the same bit in the Status port as is used for the TBMT signal described for the serial printer interface. Therefore the DMON printer driver software, (which thinks it is testing a TBMT bit to see if a UART transmit buffer is empty), in fact is testing a printer "Busy" signal to see if the printer is not busy, and there is thus no change required in the software to suit either system.

Toggle

In addition to the arrangements described above, you have manual control of the printer via the keyboard. The use of the code CTRL-P (ie holding the control key down and typing P) allows you to decide what will be printed and what will not. CTRL-P is a "toggle" function, ie it alternately enables and disables printing. If you want to pause during a long printout use CTRL-S (this is often called the "X-OFF" code) and to continue use CTRL-Q, ("X-ON"). These two codes, which are the ASCII codes for DC3 (device control 3) and DC1 (device control 1) respectively, are often used automatically during computer communication to stop and start the flow of data, so that the slower device can keep up with the faster device. The "protocol" which is thus developed is called "X-ON/X-OFF protocol". I just thought you might like to know that. Oh well, suit yourself.

The EPROM in which DMON is supplied is of a type to suit the Kemitron MZB-3 Z80A-CPU card, which we use at present at the heart of the Interak system. It is suitable for 5V single-rail operation, and of sufficient size (ie 2K or 4K) to accommodate DMON, which is a 2K program. If you are familiar with the MZB-3 card you will know there are two sets of address link settings which must be made: one is the "power on jump address", and the other is the "on board firmware address". To suit DMON both of these should be set to E000H. (The "H" here, and throughout this document means that the specified address is in hexadecimal notation, or "hex." for short).

The most important command is "B" for boot. Its function is to read the first sector of the the first track of the disk into memory and to execute it as code. This is a universal command for Interak, in that it suits all of the Interak type of diskettes without alteration or initialisation: 3.5", 5.25", 8", 35 track, 40 track, 77 track, 80 track, single density 128 bytes per sector, double density 512 bytes per sector, single sided and double sided. This is not quite as amazing as it sounds, because the function of the "B" command is merely to locate and load the first sector of the first track on the first side of the first disk drive in the system. The first track and sector are obviously fairly easy to find (it is the subsequent sectors which are more difficult, because their position varies widely as a result of the many different "formats" in vogue).

Legal

DMON finds and loads the first sector by successively switching the disk controller through all the legal modes of operation (legal for Interak that is) until it finds one which suits. If a successful load is not achieved after all the modes have been tried, the process is repeated, until either success is achieved or the whole procedure has been repeated 10 times. So as not to slow things down too much, the Interak convention is to record the "systems" tracks as single sided, whatever type of disk is in use, so DMON only has to try four combinations in all: density single and double, disk size 8" and non-8".

The data contained in the first sector is transferred to memory, but the precise address is not built into DMON. (This is because for special purposes in writing your own software as an advanced user you might legitimately want to run the program contained in that first sector in an address you choose.) Instead of using a fixed address, the contents of the sector are transferred to an address which is derived from the first byte of the sector; for example if the first byte of the sector was "AB" the address AB00H would be used, "34" would imply address 3400H, and so on. The "B" command finishes either by reporting an error, or if successful, by transferring program control to the next byte in the defined memory location (ie in the the examples quoted to AB01, or 3401).

Usually you will use the "B" command to boot up the Interak CP/M disk operating system. If this is done the contents of the program on the first sector then will be the CP/M "loader"; this proceeds to load the whole of the CP/M operating system into the appropriate place in memory, and finally transfers control to the "CCP" (Console Command Processor) part of CP/M, which in turn announces its presence by displaying its own sign on message. Of course we have organised things so that the system on a given diskette will be a suitable one for that size and type of diskette, which explains how the one simple "B" command is able to perform the apparent miracle of working with all of the Interak standard diskette types. In fact all of our implementations of CP/M are in the double density format, but for the maximum flexibility for additional purposes the DMON "B" command will boot both single and double density diskettes of all the sizes, provided they follow our conventions.

General

Even though booting up CP/M is the primary purpose of the "B" command you can see that the command is entirely general; for example it could boot up any other disk operating system (written by yourself perhaps?), or it could simply boot one single useful program - a diskette formatter for example.

Except during the initial stages of installing a fresh Interak CP/M system, one of the first automatic actions of the program will be an "INPUT" instruction from Port "FF", since this is the instruction which the Kemitron design of MZB-3 Z80A-CPU card employs to turn off the on board firmware (DMON in this case) to allow system RAM at the same address to be used. This explains how for example a 64K CP/M serial system can be achieved even though at first sight there is an EPROM in the way of the computer memory at locations E000H and above. (Other implementations of CP/M by other authors may choose to retain the on board firmware resident in the system at all times - this is not without its own merits, but of course it limits the maximum amount of available RAM to 56K - this is why we don't do it.)

If the EPROM is switched out of the memory map, the design of the Kemitron MZB-3 card does not allow it to be switched in again by software; the only way to do this is to use the CPU reset switch. This is usually no hardship, since once the diskette has been "booted" DMON has served its purpose and is not required again that session, because all programs and debugging aids etc can now be loaded from diskette.

Doomed

At power on or reset a simple test is carried out to find how much RAM is available, starting at 0000H and finishing at DFFFH, the location just before the start of DMON at E000H. The contents of the RAM are not corrupted by this test. The result is reported on the screen and lets you determine that there is sufficient memory for your purpose before using the "B" command.

(For example an attempt to run a 64K CP/M in a system which only has 20K of RAM is doomed to failure!) As DMON is initially in residence at locations E000H onwards, its RAM test only goes as high as DFFFH (the highest RAM location before you meet DMON).

DMON also contains simple disk read and write commands (for use with suitably formatted disks). Note that before these they are employed you must first set the disk parameters in memory by means of the DMON "SD" (Set Disks) command, to suit the particular size of disks you are using, the density and number of sides and which of the possible four disk drives. (This procedure was not necessary before using the "B" command, because "B" simply tries all legal formats of diskette - this is fine for a read access, but would be a disaster if writing had to take place, and in any case would be excessively slow if more than one sector was to be loaded. Invariably the "B" command itself loads and executes a program which can take control and do anything necessary).

The disk read and write commands allow reading and writing to a specific sector or sequential group of sectors you choose, starting at a specific track you choose, using memory at an address you choose. Thus for extreme economy you just about could use the DMON program alone and defer the purchase of CP/M, nevertheless enjoying the use of disks to store your programs and data. We ourselves would not think of using disks without a disk operating system like CP/M (in the same way as we would not bother to turn a television set on if there were no programs, only a test card!) but it is not inconceivable that some users might use the "B" command to boot up their own elementary disk operating system.

Although we have indicated that this would not be our idea of fun, using DMON in this way would allow you features similar to a simple ZYMON tape system but of course with vastly increased speed and reliability. Looked at in this light the use of disks without CP/M is still vastly superior to tape, but we have done our duty if we have pointed out that we ourselves would never use disks without CP/M or something similar. The disk read and write commands include basic checks for errors, and report these on screen in the form of a coded byte.

Formidable

The combination of the Interak FDC-1 disk controller, DMON monitor and Z80A-CPU running at 4 MHz without wait states is formidable, and it can accommodate the often unattainable data transfer rate of 500 kilobits per second, usually associated with 8" double density. (We allow ourselves the occasional luxury of a smirk of smug self-satisfaction when we read of highly advanced Z80 systems with all bells and whistles including DMA channels and vector driven interrupt structures, which nevertheless are not able to cope with the 500 kilobit per second transfer rate. And we can be even smuggier in noting that the designers of the magnificent 16 bit machines can't manage it either without resorting to complicated DMA operation and the like. Obviously second-division 8-bit processors such as the 6502 are similarly embarrassed, but we don't poke too much fun at them, because we're not ones to kick sand in a weakling's face.)

The data transfer rate depends on the density, and the size and rotational speed of the disk, and is as follows for a few different types of drives:

3.5" and 5.25" 300 rpm single density: 125 kilobits per second. (eg standard BBC computer)

3.5" and 5.25" 300 rpm double density: 250 kilobits per second.

3.5" 600 rpm single density: 250 kilobits per second; double density: 500 kilobits per second.

8" 360 rpm single density: 250 kilobits per second; double density: 500 kilobits per second.

Quaint

You may have noticed in the list above a particularly attractive system: the 600 rpm 3.5" drive. This has the neat size and large capacity of all 3.5" drives, together with a 100% faster data transfer rate, as fast in fact as our old favourites the 8" double density drives, ie 500 kilobits per second, ie fast! Sadly the 600 rpm drives, which as far as we know were only made by Sony, who invented the 3.5" format, have fallen out of favour. You and I know the reason why: most computers (even 16 bit) can't handle the high data rate. It is a great shame that most people only buy what fashion says is the best, not what actually is the best, and a golden opportunity to double the data transfer rate has thus passed us by. (This all does serve as a reminder that the Z80A, whilst admittedly not suitable for building IBM personal computers, nevertheless has in its quaint instruction set a few finely tuned instruction sequences that enable it to run rings round the big boys when it comes to fast data control.)

Other commands in DMON are of use in the very early stages of installation, particularly in the testing and fault finding of a newly built system which has no other software. If Interak was being built on a factory production line faults could be found before the computer left the factory, but as it is a system which you build yourself you will appreciate the extra tools in the form of DMON commands in the event ot trouble. If Interak was a motor car it would be the one which is supplied with a starting handle - we don't expect trouble, but if you do meet it you have the tools to cope!

Bonus

Once the disks are commissioned and working correctly under CP/M, the normal technique for debugging etc is to load a more comprehensive disk-orientated monitor from disk to carry out more sophisticated tasks. (One of the "free" programs included with our implementation of CP/M is the Digital Research "DDT" debugger.) However the simple monitor commands included in DMON are sufficient for the purpose, and of course a welcome bonus in a boot EPROM, which normally has no such commands.

Using DMON the contents of memory can be examined and altered, and dumped to the screen and/or printer. Also the essential port input and output instructions are included. Simple programs can be entered by hand (and stored on disk as just explained above, if need be), and run using the "G" command. Thus in the event of a new disk system not working when it is first switched on, you do have some tools available to start tracing what has gone wrong. (Whether or not you can use the tools in DMON is another matter, and depends on your own skill and experience.) Do not think that the provision of tools for roadside repairs means that you are on your own if you are in trouble, the Greenbank RAC naturally will come to your aid if you meet any difficulty in getting a system up and running for the first time. In fact this after sales service is one of the strongest advantages of the Interak system and one which we do not always remember to broadcast as LOUDLY AS WE SHOULD!

Before we leave the subject of the monitor facilities in DMON we should point out that for those users who are aligning their own floppy disk interface cards (rather than using the service we provide) will find it absolutely vital to be able to write various data to certain control ports during the setting up procedure. This is where the alternative concept fails of a simple "boot" EPROM with no other commands.

Communications

The "S" (but not the "M" and "H" versions of DMON version, which do not have the necessary serial port routines) has a powerful built in "communications" command, which allows the Interak computer and its console together to behave as the terminal for some other, usually disk based, computer. Using a simple "reversed" RS-232 lead, (sometimes called a "null modem"), the remote computer can be "milked" of its

software, even including the CP/M system itself. Note that this feature is provided only so that an existing CP/M etc licence holder can transfer legal copies of his software to his own new Interak disk machine. Note that if you do not have your own CP/M you must buy it (preferably from us!) as it is illegal to use it otherwise.

There are some temporary problems to do with the use of DMON at the moment, and we would like to draw them to your attention before you purchase. The fundamental problem is that much of the work on DMON was carried out by Mr W Schroeder, who died before he could complete all that he wanted to do. The intention was to provide in the manual for this product full details of how the various routines in DMON could be called by the user's own programs. A demonstration of the usefulness of this technique is to be had in our CP/M implementation, which begins by communicating entirely via routines within the DMON monitor (ie routines to read the keyboard, write to the screen, read disk sectors and write disk sectors, seek tracks, and so on). Of course we (or any boffin we can rope in) will in due course provide the missing information, but it is not available at the moment. This however should not detract from the attractiveness of DMON as a purchase because it is being sold at the same sort of price you could expect merely for a simple boot program; all the rest is a bonus.

Unaffected

Similarly there was an intention to provide - for the benefit of those users who cannot yet install CP/M - a disk formatting program to run under DMON. (A version of the formatter to run under CP/M is included with our CP/M, so CP/M users are unaffected.) Without a supply of formatted diskettes you cannot use the disk read and disk write commands in DMON.

We are working on the "official" solution (ie a formatting program which will run under DMON without the need for CP/M), but in the meantime we can get you over the problem in the following ways: If you purchase your diskettes from us we will format them for you free of charge before we despatch them, or, if you already have your own diskettes, we will format them for you (assuming we have the equipment) for a nominal charge, given at the end of this leaflet.

Dynamite

Note that this is not as inconvenient as it sounds, since once formatted a diskette, in theory at least, never needs formatting again. Indeed, to this day many manufacturers of word-processors and other computers do not trust their users with a formatting program (which it must be admitted is dynamite in untrained hands) and instead supply preformatted diskettes. When we eventually do have a DMON formatting program to offer it will have the same features as the one supplied with our CP/M; ie it will suit any size of disk, Interak single or double density (128 or 512 bytes per sector respectively), single or double sided, with user selectable write precompensation, and will be for use in any one of four drives, with any skew (interleaf) factor, for formatting any number of tracks, starting at any chosen track. Each track will be verified by reading the formatted data after it has been written.

LIST OF DMON COMMANDS

Select/Set:	SP	System Parameters, Disk Density, Diameter, Stepping Rate.
	SD	Select Drive, Sector, Track, Memory Address.
Disk:	DR	Disk Read (number of sectors).
	DW	Disk Write (number of sectors).
Boot:	B	Boot from disk (No need to use SP, SD etc first).
Memory:	ML	Memory Load from Keyboard (Hex. and ASCII).
	MD	Memory Dump (Hex. and ASCII) to Screen and/or Printer.
	MM	Memory Move (Copy).
	MF	Memory Fill an area with a specified byte.
	MS	Memory Search for specified byte (or pair of bytes).
Go:	G	Go (Execute) a program.
Port:	P	Input/Output directly from/to a Port - input display in both hex and binary.

(Top of RAM (if below E000H) determined automatically on reset.)

CTRL-C to regain command mode, CTRL-P to toggle printer on/off, CTRL-S (X-OFF) to stop listing, CTRL-Q (X-ON) to restart listing.

Technical Note: The disk read and write commands work with Interak standard disks of any skew whatever. No sector translation table is built into DMON so diskettes which have been recorded on other systems performing sector translation by means of internal skew tables cannot be read directly by DMON. It is only an experienced user who would be trying to read such formats anyway, and such a user would find it an almost trivial matter to write a small program to unscramble the sectors in memory if he had to read them using DMON. More than likely however our implementation of CP/M would be owned by the serious user and this provides (in the "USER" area, see the data sheet on our CP/M for more details) facilities for reading all manner of "alien" formats, with and without skew, generally by simple patching to tables in our CP/M BIOS, (not a job for a beginner, but perfectly possible once you know what you're doing).

CONTENTS OF PACKAGE

1 off DMON EPROM 24-pin 5V only EPROM to suit Kemitron MZB-3 Z80A-CPU card, with power on jump and EPROM address both set to E000H.

1 off All currently available documentation and instructions on use. A4 size, punched with 4 holes on 80 mm centres to suit Interak de-luxe binder.

ORDERING INFORMATION, PRICES

Order as "DMON-M" for standard Interak system using LKP-1 parallel keyboard interface and VDU-2K memory mapped video interface.

Order as "DMON-H" for special Interak system using LKP-1 parallel keyboard interface and your own serial (or similar port-mapped) VDU, eg the "Intelgraph" VDU board (not available from us).

Order as "DMON-S" for special Interak system using a conventional serial VDU "terminal", ie combined VDU screen and keyboard, both addressed at the same I/O ports.

Price (each type) . . . . . 14.95 + VAT

Formatted diskettes for use with DMON

As mentioned in the above description, a diskette formatting program to run under DMON is not available at present. (If you have our CP/M a formatter is provided, so there is no problem then.) In the meantime we offer a disk formatting service for DMON users.

Note that we can only format the following types of diskettes at present (but pleas enquire for other types, because we may know someone who can format them):

3.5" 80 track, double or single sided, double or single density.

5.25" 80 track, double or single sided, double or single density.

8" single sided, double or single density.

Formatting free on request if we supply the diskettes, or 50p per diskette + postage (+ VAT) on your own diskettes.